Homework 2: Q2

Name: Waiwai Kim, waiwaiki

1 Part (a) Proof Idea

Part (a) asks us to show that a family of inputs exists such that a change of a man's preference list results in a set of stable marriages different from the original set of stable marriages. In the recitation, we have learned a specific example of $n = 3$ as shown in Table 1. Table 1 results in a set of (m_1, w_1), (m_2, w_2), (m_3, w_3). On the other hand, running the GS algorithm on Table 2 after m_3 changes his preference lists results in (m_1, w_1), (m_2, w_3), (m_3, w_2). Note that the two pairs except (m_1, w_1) are changed.

Table 1: Preference Lists Prior to a Change

m1:	w1	w2	w3	w1:	m1	m2	m3
m2:	w2	w3	w1	w2:	m1	m3	m2
m3:	w3	w1	w2	w3:	m3	m2	m1

Table 2: Preference Lists After m_3's Change

m1:	w1	w2	w3	w1:	m1	m2	m3
m2:	w2	w3	w1	w2:	m1	m3	m2
m3:	w2	w3	w1	w3:	m3	m2	m1

To prove that such case exists for a general input size n is to simply keep the structure shown in Table 1 and append 4 to n so that m_i is matched with w_i. In this structure, m_i always remains with w_i where $i > 3$ regardless of m_3's preference list. In other words, running the GS algorithm on Table 3 results in a set of (m_1, w_1), (m_2, w_2), (m_3, w_3) and (m_i, w_i), where $3 < i \le n$. Call this set A. On the other hand, running the GS algorithm on Table 4 results in a set of (m_1, w_1), (m_2, w_3), (m_3, w_2) and (m_i, w_i) where $3 < i \le n$. Call this set B. Note that $A \cap B \neq A \cup B$ because A has (m_2, w_2) and (m_3, w_3) while B has (m_2, w_3) and (m_3, w_2). This is possible because having two pairs of marriages swap their respective partners among total n pairs of marriages is a sufficient condition for our proof.

Table 3: Generalized version of Table 1

m_1:	w_1	w_2	w_3	w_4	w_5	…	w_{n-1}	w_n	w_1:	m_1	m_2	m_3	m_4	m_5	…	m_{n-1}	m_n
m_2:	w_2	w_3	w_1	w_4	w_5	…	w_{n-1}	w_n	w_2:	m_1	m_3	m_2	m_4	m_5	…	m_{n-1}	m_n
m_3:	w_3	w_1	w_2	w_4	w_5	…	w_{n-1}	w_n	w_3:	m_3	m_2	m_1	m_4	m_5	…	m_{n-1}	m_n
m_4	w_4	w_5	w_6	w_7	w_8	…	w_2	w_3	w_4	m_4	m_5	m_6	m_7	m_8	…	m_2	m_3
…									…								
m_i	w_i	w_{i+1}		…	w_n	…	w_{i-2}	w_{i-1}	w_i	m_i	m_{i+1}		…	m_n	…	m_{i-2}	m_{i-1}
…									…								
m_n	w_n	w_1	w_2		..		w_{n-2}	w_{n-1}	w_n	m_n	m_1	m_2		..		m_{n-2}	m_{n-1}

Table 4: Generalized version of Table 2 after a change

m_1:	w_1	w_2	w_3	w_4	w_5	...	w_{n-1}	w_n
m_2:	w_2	w_3	w_1	w_4	w_5	...	w_{n-1}	w_n
m_3:	w_2	w_3	w_1	w_4	w_5	...	w_{n-1}	w_n
m_4	w_4	w_5	w_6	w_7	w_8	...	w_2	w_3
...								
m_i	w_i	w_{i+1}		...	w_n	...	w_{i-2}	w_{i-1}
...								
m_n	w_n	w_1	w_2		..		w_{n-2}	w_{n-1}

w_1:	m_1	m_2	m_3	m_4	m_5	...	m_{n-1}	m_n
w_2:	m_1	m_3	m_2	m_4	m_5	...	m_{n-1}	m_n
w_3:	m_3	m_2	m_1	m_4	m_5	...	m_{n-1}	m_n
w_4	m_4	m_5	m_6	m_7	m_8	...	m_2	m_3
...								
w_i	m_i	m_{i+1}		...	m_n	...	m_{i-2}	m_{i-1}
...								
w_n	m_n	m_1	m_2		..		m_{n-2}	m_{n-1}

2 Part (b) Proof Idea

The idea to trigger a swapping of two pairs in Part (a) can be extended to induce a home-wrecker in which every pair is altered after a change of one man's preference. We will show an instance that changing m_1's preference in Figure 1 will result in Figure 2.

Figure 1: Graphical Example of Home-Wrecker: Prior to Change

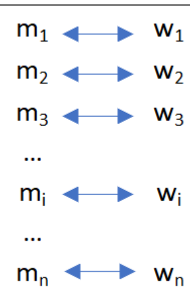
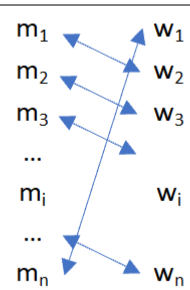


Figure 2: Graphical Example of Home-Wrecker: After to Change



For example, Let's run the GS algorithm on Table 5. The very first couple of (m_1, w_1) is formed. Shortly after, w_2 proposes to m_1 who is already matched w_1 and rejects w_2 because he prefers w_1 to w_2. w_2 continues to propose to her second favorite man m_2 and is matched with him. This pattern of proposing, getting rejected and being matched with the second favorite man repeats until the end where $n = 4$. Table 5 is synonymous to Figure 1.

On the other hand, in Table 6, m_1 changes his mind and now prefers w_2 to w_1. The very first couple of (m_1, w_1) is formed. Shortly after, w_2 proposes to m_1 who is already matched w_1 and accepts w_2 because he prefers w_2 to w_1. w_1 becomes single. It is important to note that in our algorithm that w_1 has to wait until the rest of women has a chance to propose once. w_3 proposes to m_2 and gets matched with him because he is free. Likewise (m_3, m_4) is formed.

Lastly, w_1 proposes to m_2 and m_3; however, both men already have partners and prefer their current partners to w_1. Eventually, w_1 is matched with m_4 because he is the only free man. Table 6 is synonymous to Figure 2.

Table 5: $n = 4$ Home-Wrecker Instance - Prior to Change

m1:	1	2	3	4	w1:	1	2	3	4
m2:	2	3	4	1	w2:	1	2	3	4
m3:	3	4	1	2	w3:	2	3	4	1
m4:	4	1	2	3	w4:	3	4	1	2

Table 6: $n = 4$ Home-Wrecker Instance - After Change

m1:	2	1	3	4	w1:	1	2	3	4
m2:	2	3	4	1	w2:	1	2	3	4
m3:	3	4	1	2	w3:	2	3	4	1
m4:	4	1	2	3	w4:	3	4	1	2

3 Part (b) Proof Details

If you generalize Table 5, you get Table 7. Note that m_1 is (1 , 2, 3 ... n) and subsequent man's preference is shifted one left with wraparound as i is incremented. w_1 and w_2 are (1 , 2, 3 ... n) and subsequent woman's preference is shifted one left with wraparound as i is incremented.

In Table 7, w_1 and m_1 is matched and stable. Consider (w_1, m_1) as the base case. Subsequently, w_i always proposes to m_{n-1} first; however, m_{n-1} is already matched with w_{n-1} and he prefers w_{n-1} to w_i. After getting rejected, w_i proposes to m_i, and is matched with m_i. The output from running the GS algorithm on Table 7 can be expressed as Equation 1.

$$S_{original} = \{(m_i, w_i) | 1 \leq i \leq n\} \tag{1}$$

Table 7: Generalized Home-Wrecker - Prior to Change

m_1:	1	2	3	...	i	...	n	w_1:	1	2	3	...	i	...	n
m_2:	2	3	4	...	i+1	...	1	w_2:	1	2	3	...	i	...	n
m_3:	3	4	1	...	i+2	...	2	w_3:	2	3	4	...	i+1	...	1
...								...							
m_i	i	i+1	i+2	...		...	i-1	w_i:	i-1	i	i+1	...		...	i-2
...								...							
m_n:	n	1	2	...		...	n-1	w_n:	n-1	n	n+1			1	2

If you generalize Table 6, you get Table 8. Note that m_1 is (2 , 1, 3 ... n), m_2 is (2, 3, 4 ... n, 1), and subsequent man's preference is shifted one left with wraparound as i is incremented. w_1 and w_2 are (1 , 2, 3 ... n) and subsequent woman's preference is shifted one left with wraparound as i is incremented.

In Table 8, w_1 is free as a result of w_2 proposing to m_1 w_2 is matched with m_1. Consider (m_1,w_2) as the base case. Subsequently, w_i always proposes to m_{i-1} and is matched with him on the first try because m_{i-1} is free. After (m_{n-1},

w_n) is formed w_1 begins proposing form m_2 to m_n. Note that w_1's index in m_i's preference list is $n-i+2$. On the other hand, the index of m_i's current partner is always 1. It is true that $n-i+2>1$ for $1 \leq i \leq n$. This means all m_i except m_n who is free prefers their current partners to w_1. Thus, w_1 is matched with m_n. The output from running the GS algorithm on Table 8 can be expressed as Equation 2.

$$S_{new} = \{(m_n, w_1), (m_{i-1}, w_i) | 2 \leq i \leq n\} \tag{2}$$

Based on the Equation 1 and 2, we can argue that $S_{original}$ and S_{new} are disjoint as Equation 3 shows. Equation 3 can be proven by contradiction. Assume that an element of $S_{original}$ can be found in S_{new}. The element of $S_{original}$ has m_j and w_k where $j = k$. This contradicts the condition to be an element of S_{new} where (m_l, w_k) and $l \neq k$. This contradicts the assumption. Thus it must be true that $S_{original}$ and S_{new} are disjoint.

$$S_{original} \cap S_{new} \equiv \emptyset \tag{3}$$

Table 8: Generalized Home-Wrecker - After Change

m_1:	2	1	3	…	i	…	n	w_1:	1	2	3	…	i	…	n
m_2:	2	3	4	…	i+1	…	1	w_2:	1	2	3	…	i	…	n
m_3:	3	4	1	…	i+2	…	2	w_3:	2	3	4	…	i+1	…	1
…								…							
m_i	i	i+1	i+2	…		…	i-1	w_i:	i-1	i	i+1	…		…	i-2
…								…							
m_n:	n	1	2	…		…	n-1	w_n:	n-1	n	n+1			1	2